

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

**WEB OF SCIENCE™**Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com

Acquisition and Digital Images Processing, Comparative Analysis of FPGA, DSP, PC for the Subtraction and Thresholding.

Carlos Lujan Ramirez¹, Ramón Atoche Enseñat¹
and Francisco José Mora Mas²

¹*Instituto Tecnológico de Mérida*

²*Universidad Politécnica de Valencia*

¹*Mexico*

²*Spain*

1. Introduction

Given the need to reduce the time involved in image processing, we found that is necessary to use new methods to improve the time response. In our application of real time tracking of different species in the study of marine animal conduct, a reduction in the image processing time provides us with more information that allows us to better predict the animal's escape trajectory from a predator.

Nowadays, photographic cameras can deliver photographs in various formats, hence the need to provide them with pre-processing in order to deliver the required format. Although Photographic cameras are available that can deliver photos without pre-processing, the format provided is the Bayer format. A characteristic of this format is that we obtain the images almost directly from the CCD sensors and from the analogical to digital converter with no pre-processing. The only requirement is a deserialization step and registers in parallel whose role is to place the Bayer format image in a memory for the investigation (Lujan et al., 2007). In the initial method in our project, the camera is connected to a Frame Grabber which retrieves the image, converts it to RGB, changing it to a scale of greys and subsequently carries out the necessary processing, the first being the subtraction of the background from the image of the animals under study, followed by the application of blurring.

Subtraction of images is widely used when the aim is to obtain the animal's trajectory (Gonzales et al., 2002), however, we use it to leave only the animals under study, eliminating everything else. A low pass filter (Gonzales et al., 2002) (blurred) is used which unifies any division in some of the animals caused by the subtraction. Occasionally a shrimp would be divided in two due to the kind of background used, recognizing two shrimps instead of one, this error was resolved by the application of the blurring.

The new process would involve recovering the image and then to immediately carry out the image processing of subtraction and blurring, saving time by avoiding the use of a Bayer to RGB encoder and the frame grabber. We included a stage of image recovery by a number of deserializers and registers in order to deliver the image in Bayer format.

Source: Image Processing, Book edited by: Yung-Sheng Chen,
ISBN 978-953-307-026-1, pp. 572, December 2009, INTECH, Croatia, downloaded from SCIYO.COM

For this investigation, an analysis will be performed on the third part of image processing which begins after the image has been stored in an FPGA. The two processing stages analyzed will be the subtraction between the background and the image where the animals under study are found, and the subsequent thresholding from the subtraction.

This will be implemented in four different systems, and the time and resources consumed will be analyzed; first in a PC computer with a Borland C++ program, second with an embedded microprocessor programmed in C++, third with a DSP also programmed in C++, and finally with a hardware designed in VHDL.

2. Camera link digital images acquisition.

The need to acquire images at high transfer rates, it takes us to use a camera with camera link interface, and then, to save time when recovering the image; we connect the camera with FPGA instead of use a frame grabber.

2.1 The standard camera link.

The protocol camera Link (Camera Link, 2000) is a communication interface for vision applications. The interface extends the basic technology of the channel link to provide a more useful specification for vision applications.

For many years, there has been a need for a standard method for communication in the scientific and industrial digital video market. The manufacturers of frame grabbers and cameras developed products with different connections, making cable production difficult for the manufacturers and very confusing for the consumers. A standard connection between digital cameras and frame grabbers is becoming more necessary as the data transfer rate continues to rise.

Increasingly diverse cameras and advanced data transmission signals have made a standard connection such as camera Link an absolute necessity. The camera Link interface reduces both the support time and the cost of Support. The normal cable will be able to manage the increased speed of the signal and the cable assembly will allow clients to reduce their expenses.

2.2 The standard ANSI/TIA/EIA-644

The LVDS is a pattern of data transmission that uses a balanced interface and a low voltage to solve many of the problems relating to existing technologies. The lower amplitude of the signal reduces the voltage used for the line circuits, and a balance in the signals reduces the coupling noise, allowing greater transfer rates.

The LVDS, standardized in TIA/EIA-644, specifies a maximum transfer rate of 1.923 Gbps. In practice however, the maximum transfer rate is determined by the quality of the communication media between the transmitter and the line receiver. Similarly, the length and characteristics of a transmission line condition the usable transfer rate to the maximum (Cole, E., 2002)(Texas Instruments, 2002).

National Semiconductor (National Semiconductor, 2006) developed the technology of channel link as a solution for flat screens, based on LVDS for the physical layer. The technology was later extended into a method for data transmission. The channel link comprises a pair of drivers in the transmitter and in the receiver. The driver accepts 28 data signals of simple terminal and one clock signal of simple terminal. The data are transmitted in series at a rate of 7: 1 in four lines, and together with the clock they form five differential

pairs. Therefore, the receiver accepts the four pairs of data lines in LVDS plus one pair for the clock signal which will recover the 28 bits of data and the clock signal, as shown in Fig. 1.

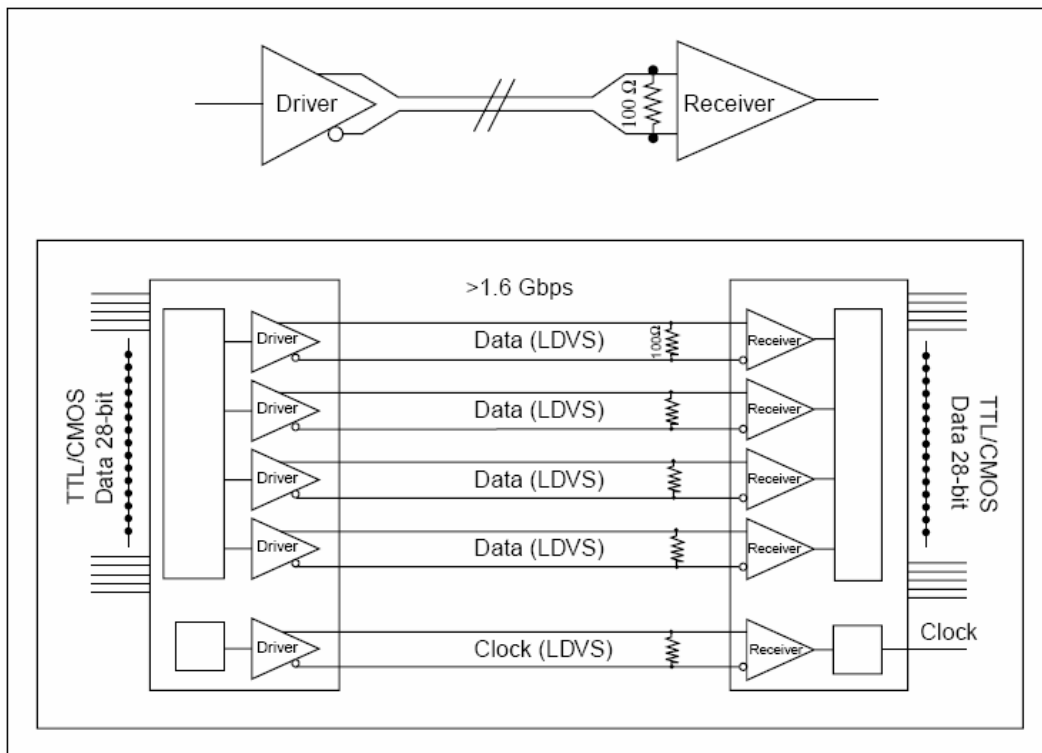


Fig. 1. Operation of channel Link

One of the benefits of the channel link transmission method is that it requires fewer conductors for transferring data. Five pairs of cables, therefore, can transmit up to 28 data bits. These cables reduce the size of the connector, allowing the manufacture of smaller cameras. Furthermore, the data transmission rate in the channel link chipset can reach 2.38 Gbits/s.

2.3 Camera connection for digital image acquisition

In the first time the camera was connected a Frame Grabber and this to the computer, we propose not to use the Frame Grabber on the other hand to use a FPGA for image recovery and storage in FPGA memory for later processing. Our advantages is a savings of Frame Grabber, because its cost is high, and on the other hand we gain the total control of the image storage.

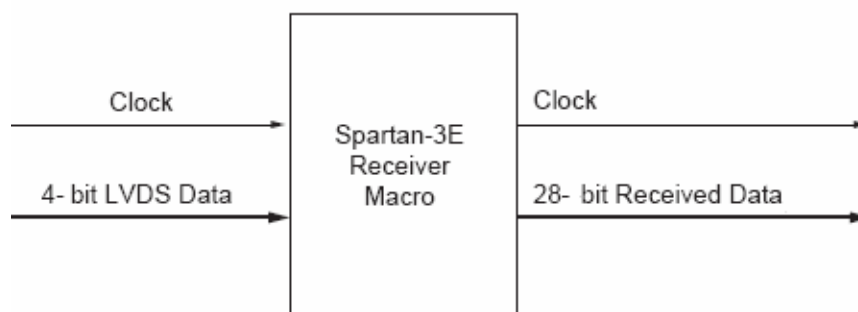


Fig. 2. Input and output receiver signals

Taking into account Sawyer implementation (2008) acquisition stage was redesigned, the FPGA was exchanged for a Virtex 4 FPGA, and a memory was added so that the image could be accessed by other devices.

The Figure 2 shows us the camera signals, which enter the receiver, and this provides us with the 28 bit data in parallel. These 28 bits are composed of 24 data bits and 4 control bits used to determine a valid pixel, an end of line and an end table. Fig. 2 shows how the camera signals enter to the receiver in serial form and the receiver gives us 28 data bits in parallel. These 28 bits are composed of 24 data bits and 4 control bits used to determine a valid pixel, an end of line and an end frame.

Logic Description

The logic for the receiver is simplified by using the cascadable IOB DDR flip-flops. The Fig. 3, shows the 4-bit interface.

Two bits of data are clocked in per line, per high-speed rxclk35 clock period, thus giving 8 bits (4-bit data width) of data to register each high-speed clock. Because this high-speed clock is a 3.5x multiple of the incoming slow clock, the bit deserialization and deframing are slightly different under the two alignment events described in "Clock Considerations." Basically, 28 bits (4-bit data width) are output each slow speed clock cycle. However, for 4-bit data width, the receiver captures 32 bits per one slow clock cycle and 24 bits the next. The decision as to which bits are output and when is made by sampling the incoming clock line and treating it as another data line. In this way, the multiplexer provides correctly deframed data to the output of the macro synchronous to the low speed clock.

There is no data buffering or flow control included in the macro, because this interface is a continuous high-speed stream and requires neither.

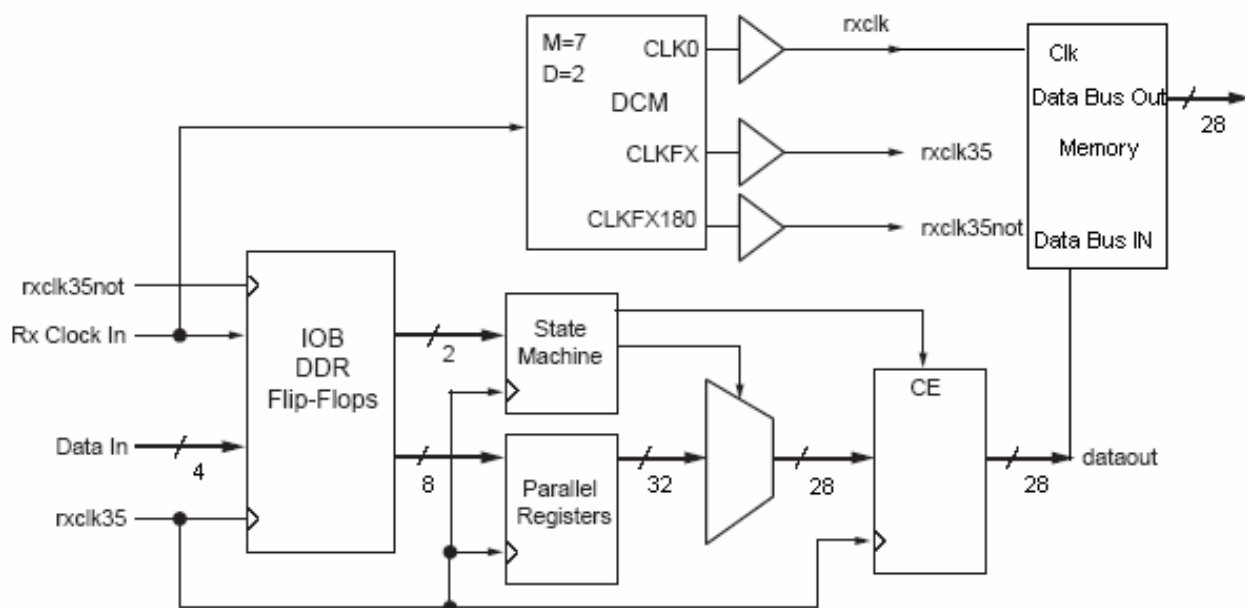


Fig. 3. Camera Link receiver implementation

It's important to be careful with the generation of the two clock signals rxclk35 and rxclk35not because the recovery of the correct images depends on them. The time diagram for these signals is shown in Fig. 4.

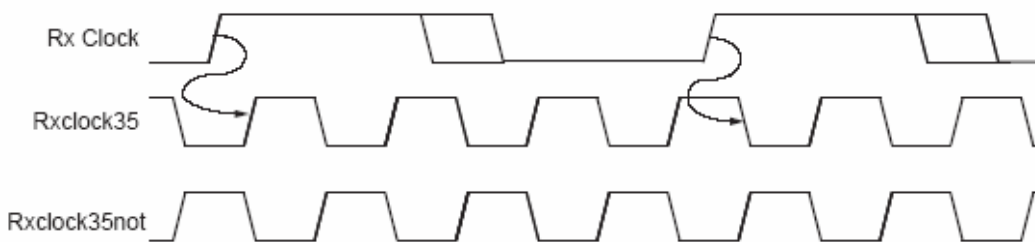


Fig. 4. Clocks wave form

The Fig. 5 shows us how the image data are transmitted through the camera link protocol.

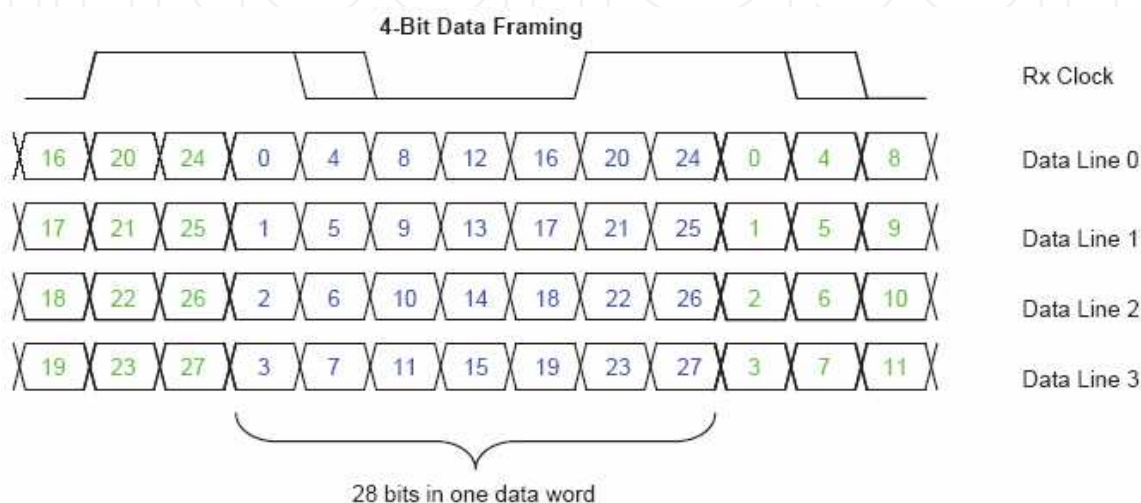


Fig. 5. Receive Data Formatting

Is very important to be careful with the placement of flip flops from deserializer, because the transfer rate of data depends of them. (Lujan et al., 2007). No use of FPGA resources (for example Digital Clock Managers in Xilixs architectures) for skew correction allows us to use in other applications that really need them.

Finally, the pixels of the digital image are stored in memory and this can be used by other devices for future processings.

In Fig. 6 shows an image of 720 x 480 pixels, recovered and stored in memory.

3. Digital images subtraction and thresholding

Two concepts that we use are defined below.

3.1 Image subtraction

The difference between two images $f(x,y)$ and $g(x,y)$ is expressed by:

$$o(i,j) = f(i,j) - g(i,j) \quad \forall i,j \tag{1}$$

This technique has several applications in segmentation and in image enhancement.

3.2 Thresholding

In order to understand this concept more clearly, we must first understand the concept of contrast amplification.

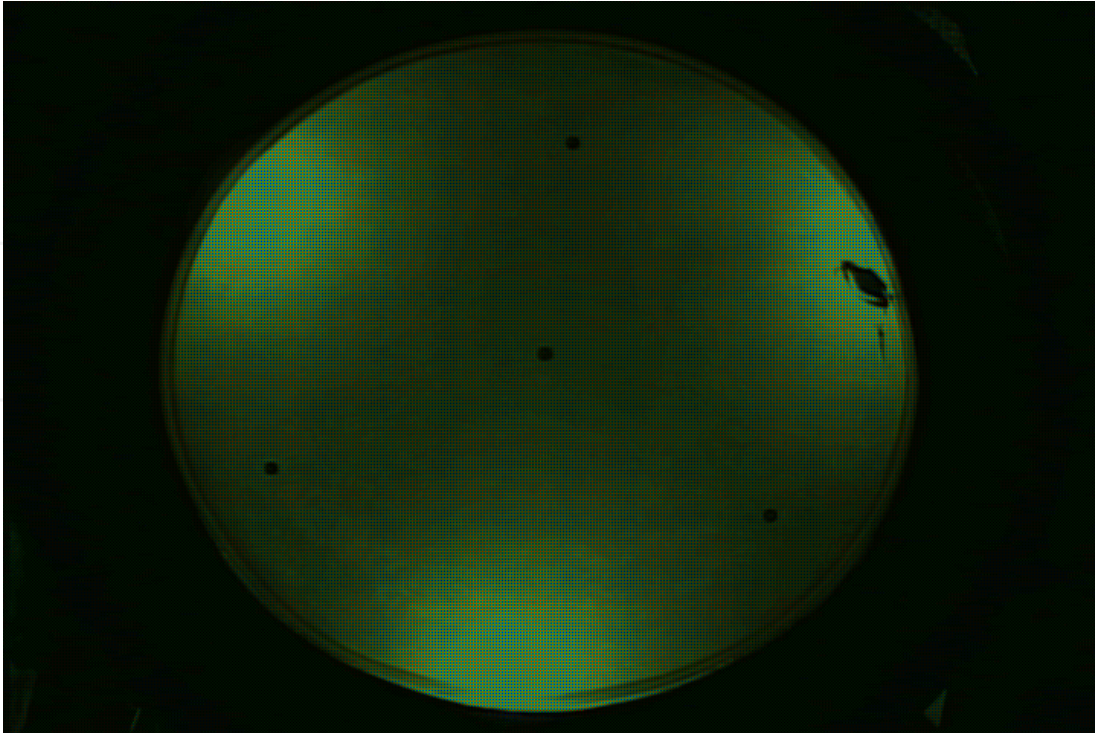


Fig. 6. Camera recovered image.

Images with poor contrast are often the result of insufficient or variable illumination, or are due to the non-linearity or small dynamic range of the image sensors. A typical example of transformation is shown in figure 7, which can be expressed by:

$$v = \begin{cases} \alpha u & 0 \leq u < a \\ \beta(u - a) + v_a & a \leq u < b \\ \gamma(u - b) + v_b & b \leq u < L \end{cases} \quad (2)$$

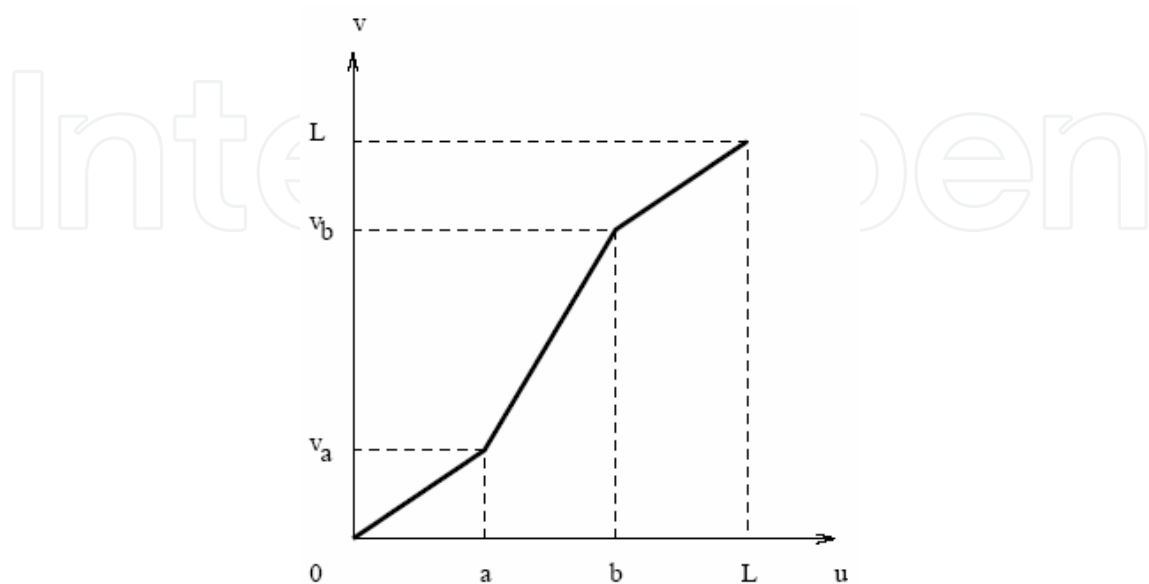


Fig. 7. Contrast amplification.

The slopes are taken greater than one in the regions in which contrast amplification is desired.

The parameters a and b can be estimated by examining the histogram of the image. For example, the intervals of grey level where the pixels occur with greater frequency must be amplified in order to enhance the visibility of the image.

A particular example of the above is when $\alpha = \gamma = 0$, this case takes the name of *cutoff*. This cutoff can be useful for instance, when we know that the signal is within the range $[a,b]$.

Thresholding is a special case of cut model in which a binary image is obtained.

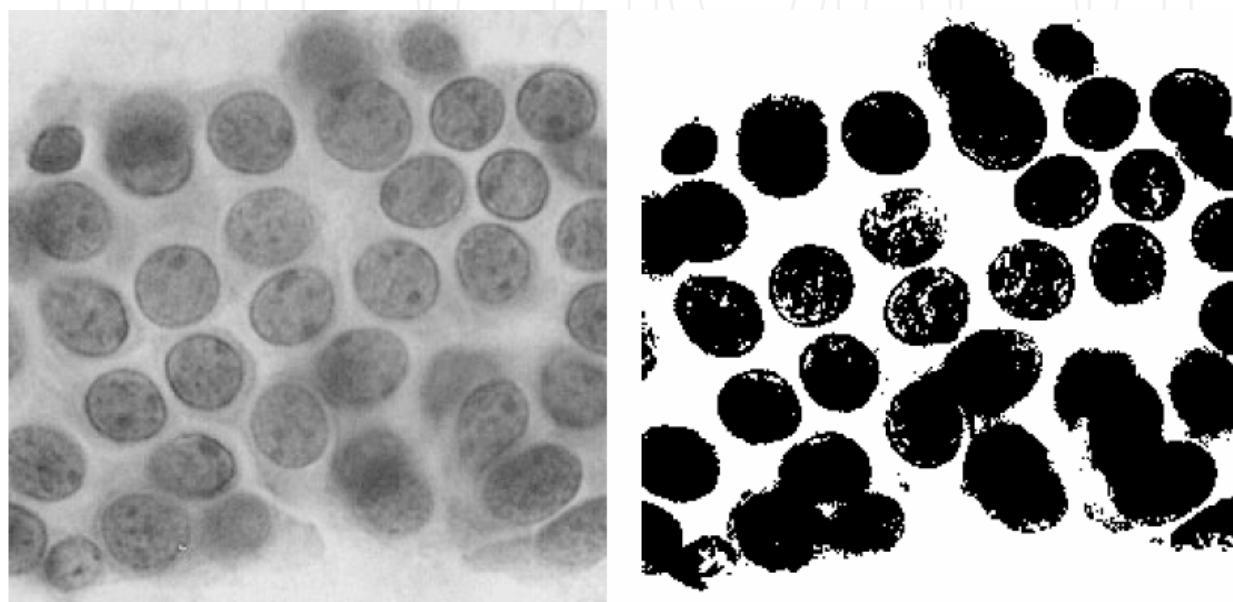


Fig. 8. Original Image and its thresholding by 150.

This can be expressed by:

$$v = \begin{cases} 0 & 0 \leq u < a \\ L & a \leq u < b \\ 0 & b \leq u < L \end{cases}$$

This model can be useful when we know that the image must be binary and when this has not been obtained by the process of digitalization. It can also be used in a segmentation process, as shown in Fig. 8.

4. Bayer format utilization for subtraction and blurring of images

4.1 Bayer format

At present, a large number of formats for storing digital images are available, however, most cameras capture the image in Bayer format (Bayer, 1976) which then passes through a converter and the image is delivered in the RGB format. The use of the Bayer color filter is a popular format for the digital acquisition of color images (National Semiconductor, 2006). A drawing of the color filters is shown in Fig. 9. Half the total number of pixels are green (G), while a quarter of the total are attributed to both red (R) and blue (B).

G	R	G	R
B	G	B	G
G	R	G	R
B	G	B	G

Fig. 9. Bayer Format.

In order to obtain the color information, the color image sensor is covered with a red, green, or blue filter, in a repetitive pattern. This pattern or sequence of the filters can vary, but the “Bayer” pattern is widely used, it is an arrangement of 2x2 repetitive and was invented by Kodak. Fig. 10 shows the positioning of the sensors and the representation of the color generated in the image.

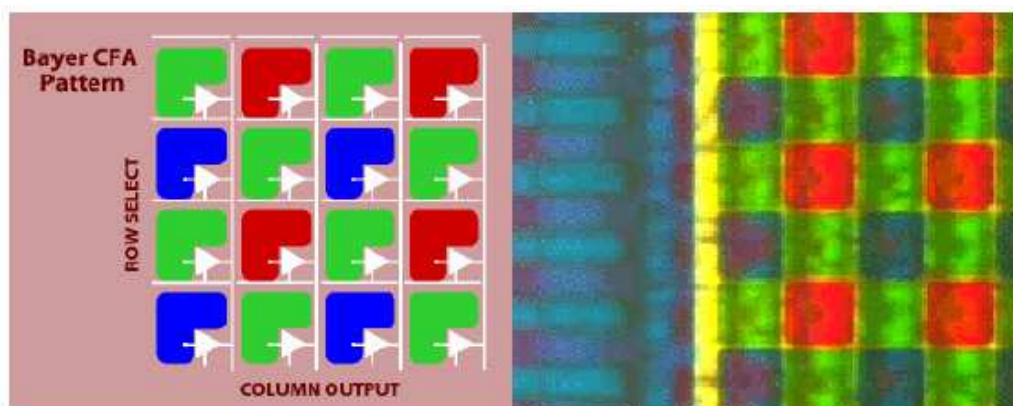


Fig. 10. Positioning of the sensors and their respective image

To obtain an image in bit map, a Bayer to RGB converter is required, there are several methods. The bilinear interpolation method together with that of linear interpolation was chosen; this correlation was taken into consideration as it presents better characteristics in comparison with others (Sakamoto et al, 1998). A brief description of the method is as follows.

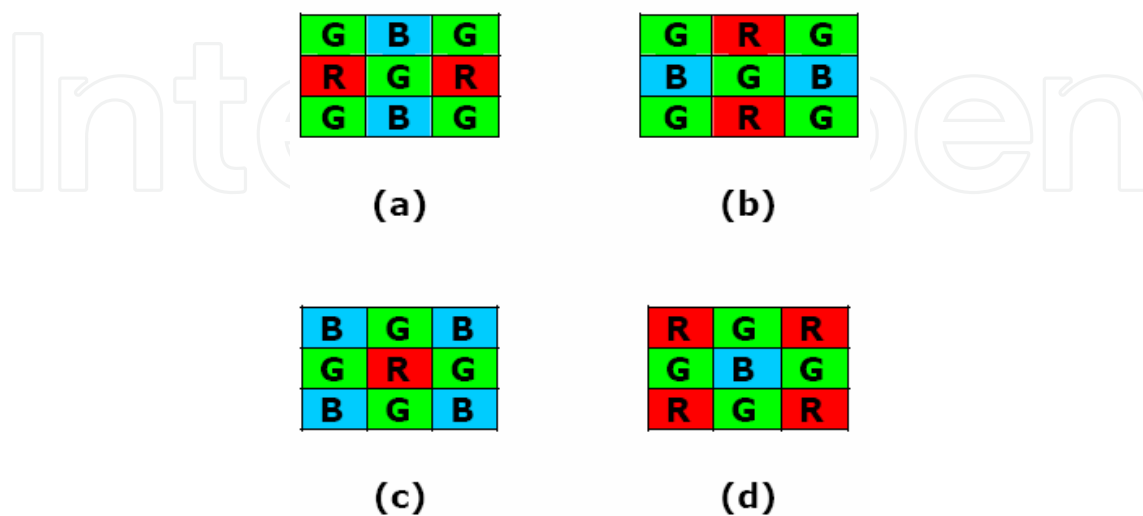


Fig. 11. Four possible cases for the interpolation of components R and B

The values of R and B are interpolated in a straight line from the closest neighbors of the same color. There are four possible cases, as shown in Fig. 11. The missing values of R and B are interpolated on a green pixel, see Fig. 11 (a) and (b), taking the mean values of the two closest neighbors with the same color. For example, in Fig. 11 (a), the value of the blue component on a G pixel will be the average of the blue pixels above and below the G pixel, while the value of the red component will be the average of the two red pixels to the left and right of the G pixel. In Fig. 11 (c) we can observe that, when the value of the blue component is interpolated for an R pixel, we take the average of the four closest blue pixels surrounding the R pixel. Similarly, to determine the value of the red component on a B pixel (Fig.11 (d)), we take the average of the four closest red pixels surrounding the B pixel.

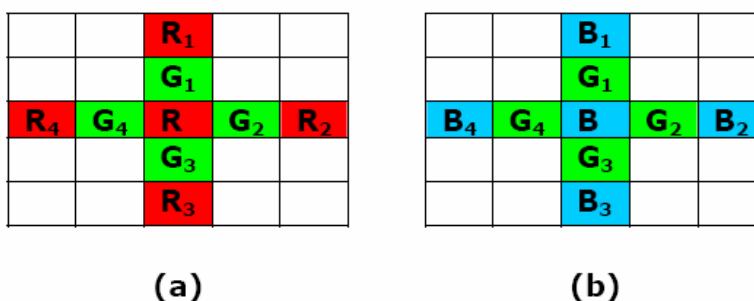


Fig. 12. two possible cases for the interpolation of the G component.

The part of the linear interpolation method dealing with the correlation is as follows: In Fig. 12(a) the value of the green component is interpolated on a R pixel. The value used for the G component is:

$$G(R) = \begin{cases} (G_1 + G_3) / 2, & \text{if } |R_1 - R_3| < |R_2 - R_4| \\ (G_2 + G_4) / 2, & \text{if } |R_1 - R_3| > |R_2 - R_4| \\ (G_1 + G_2 + G_3 + G_4) / 4, & \text{if } |R_1 - R_3| = |R_2 - R_4| \end{cases} \quad (3)$$

For Fig. 12 (b), the value of the green component is interpolated on a B pixel. The value used for the G component is as follows:

$$G(B) = \begin{cases} (G_1 + G_3) / 2, & \text{if } |B_1 - B_3| < |B_2 - B_4| \\ (G_2 + G_4) / 2, & \text{if } |B_1 - B_3| > |B_2 - B_4| \\ (G_1 + G_2 + G_3 + G_4) / 4, & \text{if } |B_1 - B_3| = |B_2 - B_4| \end{cases} \quad (4)$$

4.2 Implementation of the Bayer to RGB converter in a DSP

In accordance with the method previously described, we implemented the converter in the DSP 6416 of Texas Instruments, operating at a frequency of 1GHz (Texas Instruments, 2008). A summary of the implementation is presented in Table 1.

Size in Bytes	total of instructions	CPU cycles	time
2448	28373352	37273108	37.27 ms

Table 1. results of the implementation in DSP 6416

In the input of the converter, we have a Bayer image with the characteristics shown in Fig. 9. After applying the converter, the output would be three matrixes with the characteristics shown in Fig. 13, in other words we would obtain an image three times the size of the input.

R	R	R	R
R	R	R	R
R	R	R	R
R	R	R	R

G	G	G	G
G	G	G	G
G	G	G	G
G	G	G	G

B	B	B	B
B	B	B	B
B	B	B	B
B	B	B	B

Fig. 13. Results of the converter: RGB image

4.3 Results obtained in the tracking of the shrimp and crab.

For the following of the shrimp and crab, the first step is to subtract the background from the image of the two animals under study. The results obtained from this process using two different methods are detailed below.

4.3.1 Subtraction of images with RGB format.

First, the background of the pond must be captured (without animals). Fig. 14 shows the background and Fig. 15 shows the animals (mat), both images in RGB format.

$$p_image2=mat - background \quad (5)$$

Once we have the two images, the subtraction of images is carried out (Fig. 16) using the equation (1), after which a blurring is applied (Fig. 17).

4.3.2 Subtraction of images with Bayer format.

The procedure is the same as for the previous images, but in this case using the images in Bayer format directly before passing them through the converter. Fig. 18 shows the background image, and Fig. 19 the image with the animals.

In Fig. 20 we can observe the result of the subtraction and in Fig. 21 the result after blurring.



Fig. 14. Background with RGB format



Fig. 15. Animals in the pond with RGB format

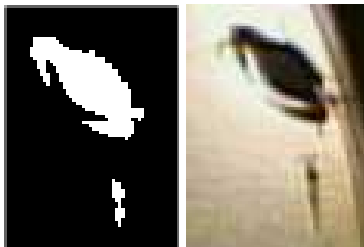


Fig. 16. Subtraction of images in RGB format

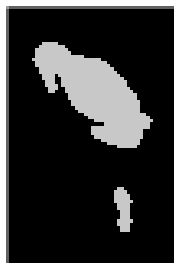


Fig. 17. Blurring of images in RGB format

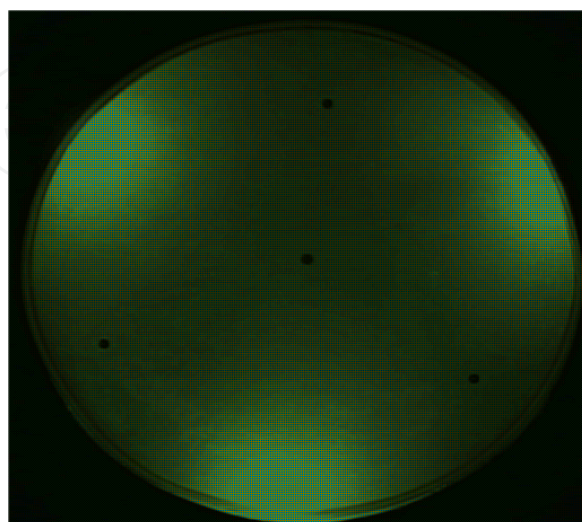


Fig. 18. Background of the pond

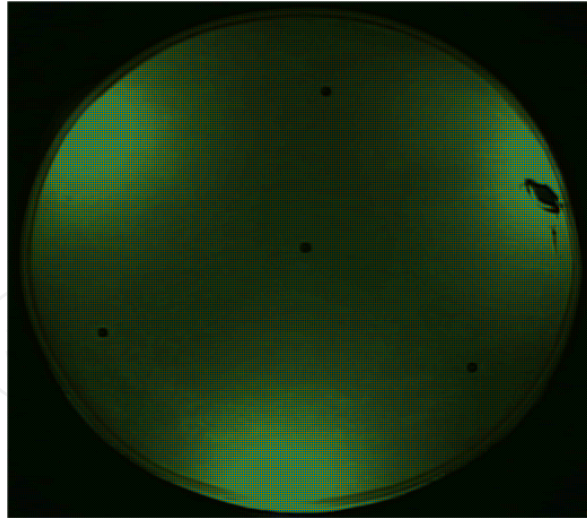


Fig. 19. Animals in the pond

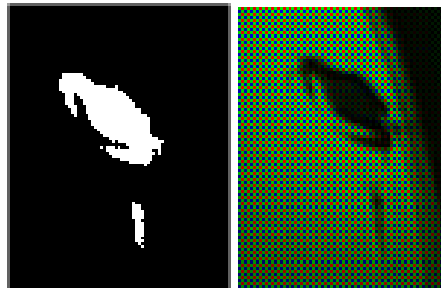


Fig. 20. Subtraction of the images in Bayer format

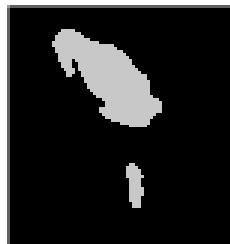


Fig. 21. Blurring of the images in Bayer format

5. Use and justification of the tools for digital images blurring and subtraction.

Once determined that is not necessary Bayer's conversion to RGB the next step is to determine the tool that it optimizes the time, to perform subtraction and blurring of the digital images.

The first objective is to separate the background from the objects to be detected. The method used to eliminate the background is that of image subtraction, in other words, the image of the scenery or background is subtracted from the images where the objects to be detected are found, obviously without the presence of the objects. The resulting image will conserve only those agents not found in the background image, which in this case will be the objects. The pixels not pertaining to the objects will remain in a range of values very close to zero and will be perceived in the image as dark or almost black colors.

The difference between two images $f(x,y)$ and $h(x,y)$, expressed as:

$$g(x,y) = f(x,y) - h(x,y) \quad (6)$$

The effect being that only the areas in which $f(x,y)$ and $h(x,y)$ are different will appear in the output image with enhanced details.

The Fig. 22, shows the images that must be subtracted in order to detect the objects, in this case the crab and the shrimp.

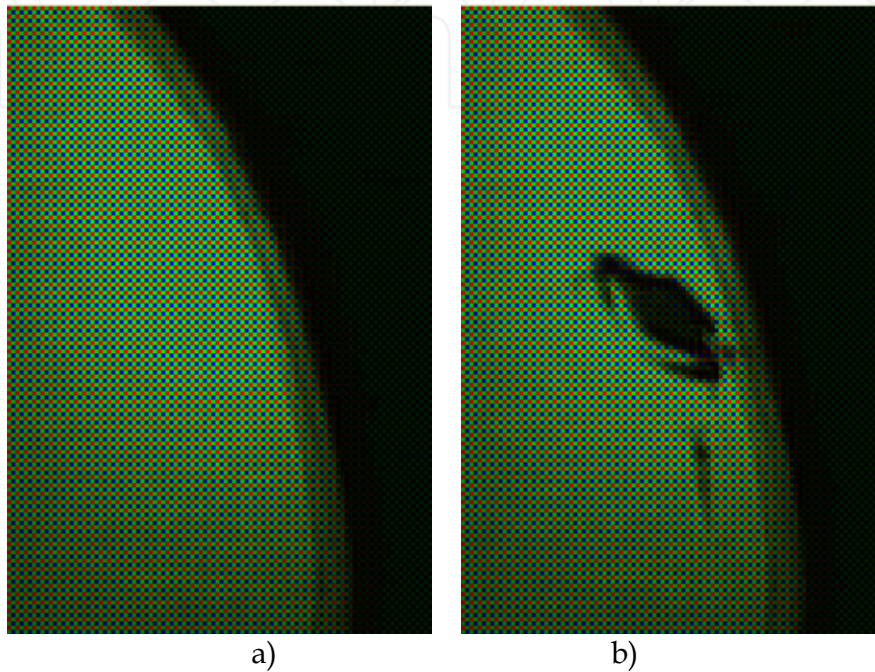


Fig. 22. In (a) we have the background image, in (b) the image with objects to be detected. The result of the subtraction can be observed in Fig. 23.

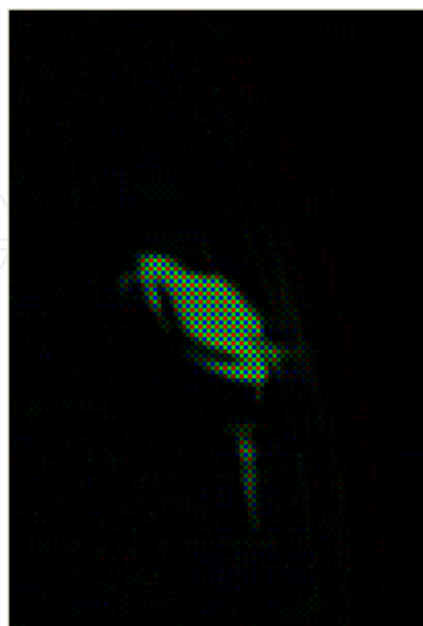


Fig. 23. Result of the subtraction.

The next step is to binarize the image by a thresholding. This is achieved by fixing a constant value of brightness, which will serve as a threshold to be able to distinguish between the sections of the image pertaining to the background and those pertaining to the objects of interest. If the brightness level of the pixel under analysis is greater than the established threshold, its value will be modified and established with the maximum brightness value in an image with an 8 bit resolution, and this value is "255". If the brightness value is not greater than the threshold, its value will be modified to the minimum brightness value corresponding to deep black, i.e. "0". The result of the thresholding can be observed in Fig. 24.



Fig. 24. Result of the thresholding

In every system in which the subtraction and thresholding were implemented, the results shown in the figures above were obtained.

Results obtained in the 4 systems implemented. For all cases the systems was implemented using an image of 102 x 150 pixels in Bayer format (Bayer, 1976)

5.1 With the Borland C++ Builder compiler in a PC.

After carrying out the implementation and executing it in the builder (2002), the next step was to measure the processing times for each process, resulting in 140 ms for the subtraction and 160 ms for the thresholding. An advantage of developing the segmentation algorithms in the Builder compiler is that they can be modified very easily and, as we are dealing with a programming language in C++, this makes the code more portable. Furthermore, a person with little knowledge of hardware development could develop the implementations, and even improve them since this tool is software orientated and an elemental knowledge of programming is sufficient. It is important to note that greater efficiency can be attained in execution times, although the main objective of using this code was its functionality rather than its efficiency. For example, predictive filtering and other techniques of algorithm development could be added. Similarly, the capacity to emigrate this code to other more

powerful programming tools, perhaps under other operating systems, could produce better results. This application was run in a computer with a 1.6 Ghz processor with which it could be possible to improve the performance by using computers with better processors.

5.2 With the Microblaze.

After downloading the project in the ML 401 board containing the “Microblaze microprocessor (Xilinx, 2002), the processing times were measured, both for the execution of the subtraction and thresholding only and for the execution of the same processes plus the blurring. In order to obtain the processing times, a flag was added to the main program which indicates when processing begins and which changes state when the execution of the segmentation algorithm ends.

The observation of the times required an oscilloscope where a testing point was placed on an output of the board and the measurement was taken. The results can be seen in Fig. 25.



Fig. 25. Processing time of the process previously described.

The duration of the width of positive pulse observed in the above figure is the total processing time, and with a tool of the oscilloscope it was possible to measure the period of the positive pulse and the exact time was obtained. The legend +Width(1)=190.0ms can be observed in the lower left-hand corner of the screen, this is the time for the subtraction and thresholding.

An advantage of developing the algorithms in the EDK is that a C++ programming language is used to program the embedded microprocessor, again making the code more portable, however, it has some instructions reserved from the Xilinx. A greater knowledge of hardware design is also required in comparison with the Builder since a hardware platform must be elaborated; however, for the user who does not have this advanced knowledge, an assistant is provided to facilitate the development of the platform

For this implementation a 100MHz master clock is used, which is the oscillating crystal of the board used for this project. The results of the algorithms presented can be improved

IntechOpen

IntechOpen

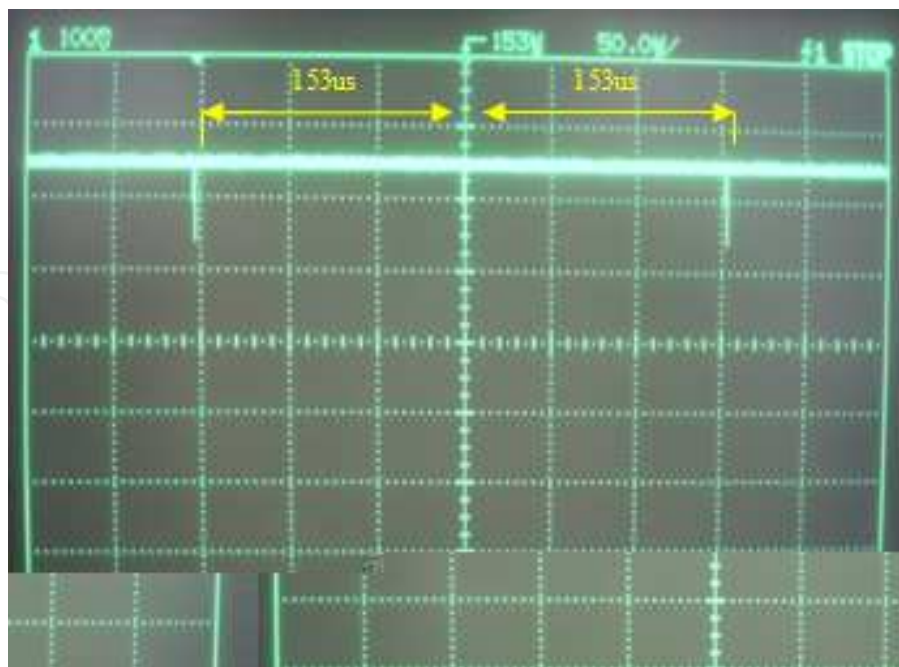


Fig. 26. Hardware processing time.

The time measured is 153us for each frame processed and is indicated in the above figure with arrows.

The results were also measured with the use of a tool within the Ise Foundation that generates a test bench with which it is possible to simulate the complete electronic system. To facilitate visualization, the necessary pines were added and, as can be observed, the processing time given is 153 us.

IntechOpen

Fig. 27. Illustration of the processing time.

In the following table the resources used for the implementation of segmentation are specified.

One of the main benefits obtained from developing the segmentation algorithms in the hardware description language (HDL) is the parallelization of the processing, with which greater frame rates would be attained since it would be possible to increase the efficiency of instructions by clock cycles.

For this implementation a 100MHz master clock is used, which is the oscillating crystal in the board used for this project. The results of the algorithms presented could be improved by using more powerful boards, such as faster clocks and boards that can handle these speeds.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	94	21,504	1%
Number of 4 input LUTs	163	21,504	1%
Logic Distribution			
Number of occupied Slices	100	10,752	1%
Number of Slices containing only related logic	100	100	100%
Number of Slices containing unrelated logic	0	100	0%
Total Number 4 input LUTs	175	21,504	1%
Number used as logic	163		
Number used as a route-thru	12		
Number of bonded IOBs	35	448	7%
Number of BUFG/BUFGCTRLs	2	32	6%
Number used as BUFGs	2		
Number used as BUFGCTRLs	0		
Number of FIFO16/RAMB16s	19	72	26%
Number used as FIFO16s	0		
Number used as RAMB16s	19		
Total equivalent gate count for design	2,193		
Additional JTAG gate count for IOBs	1,680		

Table 3. Resources used by the FPGA.

5.4 With the TMS320C6416” processor of Texas Instruments.

The subtractor and thresholder were subsequently used in the DSP 6416 of Texas Instruments, operating at a frequency of 1GHz (Texas Instruments, 2008). A summary of the implementation is described in Table 4.

Size in Bytes	Total of instructions	CPU cycles	Time
156	459754	506258	5.06 ms

Table 4. Resultados de la implementación en DSP 6416.

An advantage of developing the algorithms in the DSP is that a C++ programming language is used, which again gives more portability to the code, however, better results are obtained in comparison with the others implemented in C++, and in this case, an advanced knowledge of hardware design is not required, in contrast with the implementation in the Microblaze.

6. Conclusion

The image recovery is a fundamental part of the research, since all subsequent processes depend on it. To achieve this, departing from signals of Camera Link protocol, is very

important to have a lot of care with the location of the receiver's flip-flops, to achieve the lowest possible skew, to reach the highest transfer rate, achieving even not to use the FPGA's resources of clock managing (DCM) and to have them available for future applications.

From the obtained results, we can appreciate that converting the images to RGB format is not necessary, in other words the images obtained from the cameras in Bayer format can be used directly. Furthermore, in images 16 and 20 we can appreciate that the subtraction is better in Bayer format since the result conserves the form of the shrimp more faithfully which is of great importance as this shape will be used to obtain the orientation of the shrimp.

With this we can avoid the entire code in the implementation of the Bayer to RGB converter, and more significantly, we can save time in the conversion, which is a critical parameter in real time following of animal trajectories.

All the tests conducted on the implementations demonstrated the functionality of each one of them as well as their technological expectations.

From the results, we can see that the best results are obtained with the implementation in an FPGA, however, the complexity of programming could be a limiting factor and any change in the hardware might represent another difficulty. The recommendation therefore is to work with the mixture of technologies; in processes requiring greater speed, the FPGA can be used and in processes where this not a critical factor the DSP can be used, thereby making use of the good results obtained in this investigation. With this combination it would be possible to optimise the whole system.

7. References

- Bayer, B. E.; (1976) Color imaging array, U.S. Patent, 3971065.
- Builder 6; (2002) Borland C++ Builder 6, 2002.
- Camera Link; (2000) Specifications of the camera link, interface standard for digital camera and frame grabbers, Camera Link, October 2000.
- Cole, E.; (2002) "Performance of LVDS with Different Cables", SLLA053B, Application report, Texas Instruments, February 2002
- Gonzales, R.; Woods, R. (2002) "Digital Image Processing", Second Edition, Prentice Hall,
- Lujan, C.; Mora, F.; Martinez J. (2007) "Comparative analysis between the Stratix II (Altera) and Virtex 4 (Xilinx) for implementing a LVDS bus receiver", Electrical and Electronics Engineering, 2007. ICEEE 2007. 4th International Conference on 5-7 Sept. 2007, Page(s):373 - 376.
- National Semiconductor, (2006) Channel Link Design Guide, National semiconductor, June 2006
- Sakamoto, Nakanishi and Hase, "Software Pixel Interpolation for Digital Still Camaras Suitable for a 32-bit MCU", IEEE Transactions on Consumer Electronics, Vol. 44, No.4, November 1998.
- Sawyer N. (2008) "1:7 Deserialization in Spartan-3E/3^a FPGAs at Speeds Up to 666 Mbps", XAPP485 (v1.2), Application notes, Xilinx, 27 May 2008.
- Texas Instruments, (2002) "Interface Circuit for TIA/EIA-644 (LVDS)", SLLA038B, Application notes, Texas Instruments, September 2002.

Texas Instruments, (2008) "Fixed-Point Digital Signal Processor TMS320C6416", SPRS226K, Data Sheet, Texas Instruments, January 2008.

Xilinx, (2002) MicroBlaze Product Brief, USA, 2002.

Xilinx, (2006) ML 401/ ML 402/ ML 403 Evaluation Platform, User Guide, Xilinx, May 2006.

IntechOpen

IntechOpen



Image Processing

Edited by Yung-Sheng Chen

ISBN 978-953-307-026-1

Hard cover, 516 pages

Publisher InTech

Published online 01, December, 2009

Published in print edition December, 2009

There are six sections in this book. The first section presents basic image processing techniques, such as image acquisition, storage, retrieval, transformation, filtering, and parallel computing. Then, some applications, such as road sign recognition, air quality monitoring, remote sensed image analysis, and diagnosis of industrial parts are considered. Subsequently, the application of image processing for the special eye examination and a newly three-dimensional digital camera are introduced. On the other hand, the section of medical imaging will show the applications of nuclear imaging, ultrasound imaging, and biology. The section of neural fuzzy presents the topics of image recognition, self-learning, image restoration, as well as evolutionary. The final section will show how to implement the hardware design based on the SoC or FPGA to accelerate image processing.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Carlos Lujan Ramirez, Ramón Atoche Enseñat and Francisco José Mora Mas (2009). Acquisition and Digital Images Processing, Comparative Analysis of FPGA, DSP, PC for the Subtraction and Thresholding., Image Processing, Yung-Sheng Chen (Ed.), ISBN: 978-953-307-026-1, InTech, Available from:

<http://www.intechopen.com/books/image-processing/acquisition-and-digital-images-processing-comparative-analysis-of-fpga-dsp-pc-for-the-subtraction-an>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen